

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-182225

(43)Date of publication of application : 21.07.1995

(51)Int.Cl. G06F 12/02
G06F 15/00

(21)Application number : 05-257936

(71)Applicant : FUJITSU LTD
NIPPON TELEGR & TELEPH CORP <NTT>

(22)Date of filing : 15.10.1993

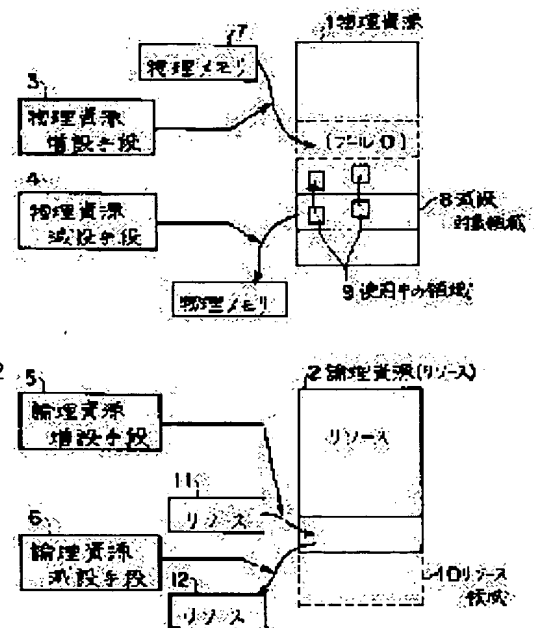
(72)Inventor : SHIBUYA KAZUO
ISHIKAWA KAZUNARI
HOSHIAI TAKANARI

(54) ON-LINE INCREASING/DECREASING SYSTEM FOR OS RESOURCE

(57)Abstract:

PURPOSE: To efficiently execute the increment/decrement of a physical resource and a logical resource through an on-line without exerting influence upon files in an OS or an application.

CONSTITUTION: In the case of increasing the physical resource 1, a physical resource increasing means 3 is started and a physical memory 7 to be increased is allocated to a pool '0'. In the case of decreasing the resource 1, a physical resource decreasing means 4 is started. The means 4 starts a monitoring task, and if an area in using exists in an area 8 to be decreased, the area is copied to an unused area and then a physical memory in the area 8 is deleted. In the case of increasing a logical resource 2, a stand-by resource area 10 for increment is previously secured, a resource 11 to be increased is allocated to the area 10 and the resource 11 is integrated into an idle chain for managing the resource. In the case of decreasing the resource 2, a resource 12 to be decreased is disconnected from the idle chain and the area of the resource 12 is integrated into the stand-by resource area 10.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-182225

(43) 公開日 平成7年(1995)7月21日

(51) Int.Cl. ⁸	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 12/02	5 1 0 A	9366-5B		
15/00	3 1 0 B	7459-5L		

審査請求 未請求 請求項の数3 O L (全 11 頁)

(21) 出願番号 特願平5-257936

(22) 出願日 平成5年(1993)10月15日

(71) 出願人 000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地

(71) 出願人 000004226
日本電信電話株式会社
東京都千代田区内幸町一丁目1番6号

(72) 発明者 渋谷 和夫
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72) 発明者 石川 一成
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74) 代理人 弁理士 大菅 義之 (外1名)

最終頁に続く

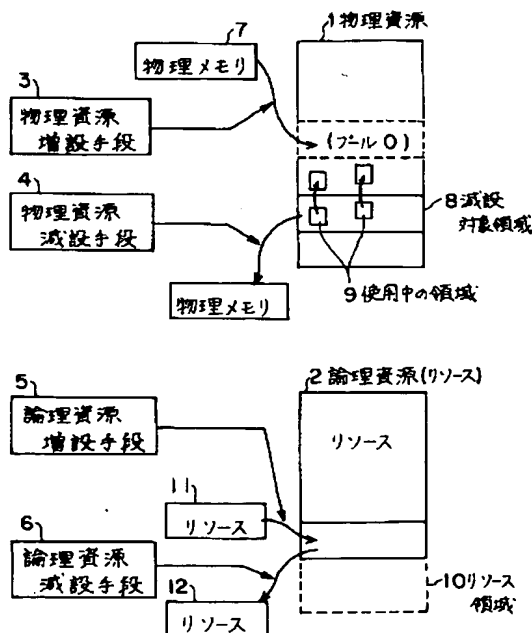
(54) 【発明の名称】 OSリソースのオンライン増減設方式

(57) 【要約】 (修正有)

【目的】 物理資源および論理資源の増減設をOSやアプリケーションのファイルに影響を及ぼすことなく、オンラインで効率的に実行する。

【構成】 物理資源1の増設を行なう場合には、物理資源増設手段3を起動し、増設する物理メモリ7をプール0に割りつける。また、減設する場合には、物理資源減設手段4を起動する。物理資源減設手段4は、監視タスクを起動し、減設対象領域8に使用中の領域があれば、該領域を未使用領域にコピーし、その後、減設対象領域8の物理メモリを削除する。論理資源(リソース)2を増設する場合には、予め増設用の予備のリソース領域10を確保しておき、リソース領域10に増設するリソース11を割り付けるとともに、リソース管理用のアイドルチェーンに該リソース11を組み入れる。一方、減設する場合には、アイドルチェーンから減設するリソース12をはずしたうえで、減設分のリソース12の領域を予備のリソース領域10に組み入れる。

本発明の機能ブロック図



【特許請求の範囲】

【請求項 1】 物理メモリ等の物理資源 1 と、OS が提供するタスク等のさまざまなサービスの状態を管理する論理資源（リソース）2 とを有するコンピュータ・システムにおいて、

オンライン処理により、増設する物理資源である物理メモリ 7 をフリー・メモリ（プール 0）に割りつける物理資源増設手段 3 と、

実行レベルの低い監視タスクを起動することにより、論理空間と物理空間の割りつけを行っているアドレス変換テーブルを検索し、減設したい物理メモリ領域 8 に使用中の領域 9 がある場合には、該使用中の領域 9 を減設したい物理メモリ領域 8 以外の物理メモリ領域にコピーしたうえ、該減設対象領域 8 の物理メモリの減設をオンライン処理で行なう物理資源減設手段 4 と、

予め論理資源（リソース）を増設するための領域であるリソース領域 10 を論理空間内に確保しておき、オンライン処理により、該リソース領域 10 中に増設分のリソース 11 を割りつけたうえ、該増設分のリソース 11 をリソースを管理するためのアイドルチェーンに組み込むことにより論理資源を増設する論理資源増設手段 5 と、オンライン処理により、該アイドルチェーンに組み込まれている減設したいリソース 12 をアイドルチェーンから解放するとともに、解放したリソース 12 のあった領域を新たに前記論理資源増設手段 5 によりリソースを増設するためのリソース領域 10 とする論理資源減設手段 6 とを、

有することを特徴とする OS リソースのオンライン増減設方式。前記物理資源増設手段 3、前記物理資源減設手段 4 は、物理資源の増減設中は、該増減設以前の物理資源の状態を保持し、障害発生時には、該増減設以前の物理資源の状態に復帰する。

【請求項 2】 請求項 1 に記載の物理資源増設手段 3 は、増設処理中には、増設以前の物理資源の情報を保持しておき、障害発生時には、増設以前の物理資源の状態に復帰することを特徴とする OS リソースのオンライン増減設方式。

【請求項 3】 請求項 1 に記載の物理資源減設手段 4 は、減設処理中には、減設以前の物理資源の情報を保持しておき、障害発生時には、減設以前の物理資源の状態に復帰することを特徴とする OS リソースのオンライン増減設方式。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 本発明は、資源の増減設を行なう場合の OS リソースのオンライン増減設方式に係り、更に詳しくは、ファイルの更新や OS の初期設定を行なうことなくオンラインで資源の増減設を行なう OS リソースのオンライン増減設方式に関する。

【0002】

【従来の技術】 OS リソースの増減設には、実装メモリ等の増減設を行なう物理資源の増減設と、OS が行なう各種サービスの状態を管理するための論理資源（リソース）の増減設がある。

【0003】 以下に、従来の物理資源の増減設方法および論理資源の増減設方法を説明する。図 11 は、物理資源を増設する場合の従来方法の説明図である。

【0004】 同図（a）に示すように、コンピュータの論理空間に OS ファイルや複数のアプリケーション・ファイル、誰もが使用可能なフリー・メモリ（プール 0）が割りつけられており、物理空間内に実際に各ファイルが格納されているとともに、フリー・メモリ（プール 0）が割りつけられている。フリー・メモリ（プール 0）はアプリケーション・ファイル（n）の後に位置しているものとする。そして、論理空間内の位置付けである論理アドレスと実際に格納されている物理空間上の位置付けである物理アドレスの対応関係がアドレス変換テーブルによって指定されている。

【0005】 例えば、メモリが不足した場合やファイルを追加したい場合には物理資源であるメモリを増設することになる。ここで、アプリケーション・ファイル（n-1）とアプリケーション・ファイル（n）の間の、今までメモリが実装されていなかった空間にメモリを実装することを考える（増設対象部分）。

【0006】 従来、フリー・メモリ（プール 0）は連続的な領域を取ることになっているので、物理空間上の該増設対象部分にメモリを実装したとしても、すぐにこの領域を利用することはできない。そのため、このままでは増設メモリが無駄になり、増設の意味がなくなる。そこで、ファイルの再構成が必要になる。すなわち、同図（b）に示すように、増設部分にアプリケーション・ファイル（n）を載せなおして、フリー・メモリ（プール 0）の領域を増設メモリ分だけ増やす作業を行なう必要がある。

【0007】 図 12 は、物理資源を減設する場合の従来方法の説明図である。ファイルが同図（a）に示すような構成で割りつけられているときに、フリー・メモリ（プール 0）の一部を減設する場合を考える。

【0008】 フリー・メモリ（プール 0）の部分はその時点で誰が使用しているかは全く分からない。そのため、減設対象部分を単純にはずすと、減設対象部分内のメモリを使用していたユーザの仕事が停止し、該メモリ内のデータ等も消失してしまうことになり、減設対象部分を単純にはずすことはできない。そこで、OS が管理している管理データの変更を行なうとともに、OS の初期設定をしない必要がある。

【0009】 図 13 は、従来の物理資源の増減設処理のフローチャートである。物理資源の増減設を開始すると（S131）、まず、既存の OS（旧 OS）上にあるハードウェア（物理資源）の実装状況に関するデータを増

減設に合わせて変更し、新しいOSファイルを作成する(S132)。次に、ハードウェアの増減設を実際に行なう(S133)。すなわち、増設メモリを実装したり、あるいは、既に実装してあるメモリを取り去る。そして、新OSファイルの初期設定(立ち上げ)を行ない(S134)、うまく立ち上がれば(OK)、新OSファイルと旧OSファイルの乗せ換えを行って、物理資源の増減設処理を終了する(S136)。うまく立ち上がらない場合(NG)は、旧OSファイルへ復帰し(S135)、一応物理資源の増減設処理を終了する(S136)。このような場合、立ち上げがうまくいかない原因を除去したうえで、もう一度物理資源の増減設処理を行なう。

【0010】次に、論理資源の増減設の従来方法について説明する。図14、図15は、論理資源を増設する場合の従来方法の説明図である。タスクやメモリ等のOSがサービスを提供するものはリソースという論理資源を使用してOSが管理している。リソースはOSが提供するサービスごとに設けられ、個々のサービスの状態などの情報が書き込まれている。

【0011】図14の(a)に示すように、論理空間および物理空間にリソース域が設けられ、合計 $n+4$ 個のリソースが割りつけられている。すなわち、このシステムでは最高 $n+4$ 個のサービスが提供可能である。そして、OSは、立ち上げ時に $n+4$ 個のリソースをチェーン状につないだアイドルチェーンを形成し、サービスを使用する場合に、このアイドルチェーンからリソースを一つはずし、それを該サービスに割り当てて状態等を書き込み、該サービスを管理する。

【0012】図14の(a)のように、論理空間と物理空間には、リソース域の他にOSファイルやアプリケーション・ファイルが割りつけられているものとする。この状態で、論理資源を増設する(リソースの数を増やす)場合を考える。例えば、5個リソースを増設するものとする。

【0013】単にリソース域を拡張するとOSファイルの領域に侵入してOSファイルを破壊することになる。そのため、ファイル全部の構成をしなおし、同図(b)に示すように、リソース $n+4$ の後にリソース5個分の領域を空け、そこに5個のリソースを挿入する。そのため、全ファイルを構成しなおすとともに、OSが管理する論理資源のデータを変更する必要がある。

【0014】図14の(a)では、リソース域を論理空間および物理空間の一番最初においたので、リソースを増設するときには全ファイルを構成しなおす必要があった。リソース域が他の位置に割りつけられていた場合にリソースを増設するときには、それ以降のファイルを構成しなおすことになる。

【0015】図16、図17は、論理資源の減設の従来方法の説明図である。 $n+9$ 個あったリソースを5個減

設する場合を例にしている。単純に $n+5 \sim n+9$ 番目のリソースを減らすと、その部分のメモリが空いて無駄になるとともに、使用中のリソースを削除する可能性がある。そこで、削除対象部分のリソースで使用しているものを他のリソースに移し替えるとともに、OSファイル以下の全ファイルを構成しなおして図16の(b)に示すように、 $n+5 \sim n+9$ 番目のリソースの領域をつめる処理を行なう。

【0016】図18は、論理資源の増減設をする場合の従来の方法のフローチャートである。論理資源の増減設を開始すると(S161)、まず、既存のOS(旧OS)上にある論理資源を管理するためのデータを増減設に合わせて変更し、新しいOSファイルおよびアプリケーション等の新ファイルを作成する(S162)。次に、新OSファイルの初期設定(立ち上げ)を行ない(S163)、うまく立ち上がれば(OK)、アイドルチェーンへのリソースの組み込み(増設)あるいは解除(減設)を行い、また新OSファイルを乗せ換えて論理資源の増減設処理を終了する(S165)。うまく立ち上がらない場合(NG)は、旧OSファイルへ復帰し(S164)、一応論理資源の増減設処理を終了する(S165)。このような場合、立ち上げがうまくいかない原因を除去したうえで、もう一度論理資源の増減設処理を行なう。

【0017】稼働中の交換機のように、運用を停止できないシステムで論理資源の増減設を行なう場合には、運用と新しいファイルの立ち上げを同時に行ない、新しいファイルの立ち上げがうまくいけば、アイドルチェーンへのリソースの組み込み(増設)あるいは解除(減設)を行ない、新OSファイルを乗せて新ファイルでの運用を開始する。

【0018】

【発明が解決しようとする課題】しかしながら、従来の方式では、物理資源や論理資源の増減設にともなってOSファイルを変更したり、ファイルを乗せ替える必要があるうえ、OSの初期設定(立ち上げ)を一から行わないとならず、時間や労力がかかるという問題があった。また、システムを運用中の場合には、運用しながらOSファイルの変更や乗せ替え、初期設定を行わなくてはならない。増減設処理中に誤りが生じた場合には運用中のシステムにも問題を及ぼす可能性があり、問題である。

【0019】本発明は、物理資源や論理資源の増減設を、システム運用中であっても、運用中の処理に影響を及ぼすことなく、短時間に、手間をかけずに行なうことを目的とする。

【0020】

【課題を解決するための手段】第1図は、本発明の機能ブロック図である。本発明は、物理メモリ等の物理資源1と、OSが提供するタスク等のさまざまなサービスの状態を管理する論理資源(リソース)2とを有するコン

ピュータ・システムを前提とする。

【0021】まず、物理資源増設手段3を有する。物理資源増設手段3は、増設する物理資源である物理メモリ7をフリー・メモリ（プール0）に割りつける。次に、物理資源減設手段4を有する。物理資源減設手段4は、実行レベルの低い監視タスクによってアドレス変換テーブルを検索し、減設対象の物理メモリ領域8が使用中の場合には、該使用中の領域9を減設対象領域8以外の物理メモリ領域にコピーしたうえ、該減設対象領域8の物理メモリの減設を行なう。

【0022】さらに、論理資源増設手段5を有する。論理資源増設手段5は、予め論理資源（リソース）を増設するためのリソース領域10を論理空間内に確保しておく、該リソース領域10中に増設分のリソース11を割りつけたうえ、該増設分のリソース11をアイドルチェーンに組み込むことにより論理資源を増設する。

【0023】最後に、論理資源減設手段6を有する。論理資源減設手段6は、アイドルチェーンに組み込まれている減設対象のリソース12をアイドルチェーンから解放するとともに、解放したリソース12のあった領域を新たに前記論理資源増設手段5によりリソースを増設するためのリソース領域10とする。

【0024】前記物理資源増設手段3、前記物理資源減設手段4は、物理資源の増減設中は、該増減設以前の物理資源の状態を保持し、障害発生時には、該増減設以前の物理資源の状態に復帰する。

【0025】

【作用】次に、図1の機能ブロック図の作用を説明する。物理資源1の空き領域に物理メモリ7を増設する場合には、物理資源増設手段3を起動する。物理資源増設手段3は、物理メモリ7を実装し、その領域をフリー・メモリ（プール0）として論理空間上および物理空間上に割りつける。これにより、OSやアプリケーション等の他のファイルに影響を及ぼさず、物理メモリ7を増設することができる。物理資源増設中は、増設以前の論理空間および物理空間上の割りつけ情報を保持しておく、障害が発生した場合には、増設以前の状態に復帰させるようにする。

【0026】また、物理資源1から一部の物理メモリを減設する場合には、物理資源減設手段4を起動する。物理資源減設手段4は、実行レベルの低い監視タスクを起動し、減設対象領域8に使用中の領域9があるか否かをアドレス変換テーブルを検索することにより判断し、存在する場合には、該使用中の領域9を減設対象領域8以外の領域にコピーする。そのうえで、物理資源減設手段4は、減設対象領域8の物理メモリをとりはずす。

【0027】一方、OSが提供するメモリやタスク等のサービスの状態を管理する論理資源（リソース）2を増設する場合には、論理資源増設手段5を起動する。論理資源（リソース）2を増設する場合を考慮して、予めリ

ソースを増設する領域（リソース領域）10を物理空間および論理空間上に確保しておく。そして、このリソース領域に増設分のリソース11を割りつけるとともに、OSが管理するリソースのアイドルチェーンに増設分のリソース11を組み込む。これにより、OSは増設分のリソース11を管理することが可能になる。

【0028】論理資源を減設する場合には論理資源減設手段6を起動する。論理資源減設手段6は、減設対象のリソース12をアイドルチェーンからはずすとともに、リソース12の論理空間および物理空間上での割りつけを解放する。そして、解放した領域を今後リソースを増設するためのリソース領域10として割りつける。

【0029】

【実施例】図2は、一実施例のシステム構成図である。本発明は、例えば、交換機等に使用するコンピュータ・システム200を前提とする。そして、コンピュータ・システム200には、キーボードやディスプレイ等のI/O装置280や、磁気ディスク等の外部記憶装置290が接続されている。

【0030】コンピュータ・システム200は、CPU210と、I/Oインタフェース230、主記憶220等で構成され、I/O装置280や外部記憶装置290はI/Oインタフェース230を介してコンピュータ・システム200に接続されている。

【0031】主記憶220は例えば何枚かの複数個のRAMを載せたメモリボード240で構成される。コンピュータ・システム200内には該メモリボード240を実装するフォルダがあり、該メモリボード240が実装されているか否かを示すテーブルとしてハードウェア実装表250がある。このハードウェア実装表250はOS221が管理する。そのほか、論理空間と物理空間のアドレスを変換するためのテーブルであるアドレス変換テーブル260を有する。

【0032】主記憶220内には、該コンピュータ・システムの管理を行なうためのOS221やOS221が提供する各種のタスク223、各種のアプリケーション・ファイル1～n（225-1～225-n）、誰もが使用できるメモリ領域であるフリー・メモリ（プール0）226等が存在する。

【0033】本発明の物理資源や論理資源の増減設を行なうためのタスクとして、増減設タスク223-2を主記憶上に載せる。物理資源や論理資源の増減設を行なう場合には、この増減設タスク223-2を起動し、処理を行なう。

【0034】増減設タスク223-2が起動されると、まず、増減設中であることを示すフラグ（増減設中指示フラグ224）を立てる。このフラグ224は、障害が発生した場合に、増減設中であるか否かを判別するために使用される。

【0035】図3は、一実施例の物理資源の増設方法の

説明図である。OSファイルやアプリケーション・ファイル1～n、フリー・メモリ（プール0）が論理空間および物理空間上で同図（a）のように割りつけられており、物理空間上でアプリケーション・ファイルnの前にメモリを増設するものとする。

【0036】本実施例の物理資源の増設方法では、増設するメモリをフリー・メモリ（プール0）に組み込むことが特徴である。図4は、一実施例の物理資源増設のフローチャートである。

【0037】まず、増減設中であることを示すフラグ（通常運転中／増減設中指示フラグ）290を立てる（S41）。そして、OSの管理データを別のメモリ領域に退避する（S42）。これはシステムが稼働している場合には、例えば交換機等の運用を停止することはできない。そこで、このフラグ290を立てたうえでデータを保持しておき、増減設中に障害が発生した場合には退避しておいたデータを元の管理データの領域に戻せるようにしておく。

【0038】そのうえで、増設メモリを実装する（S43）。次に、ハードウェア実装表の内容を変更する（S44）。すなわち、ハードウェア実装表のなかの増設メモリの位置のデータを実装済みを示す値に変更する。次に、増設メモリをフリー・メモリ（プール0）に組み込むようにOSの管理データを変更する（S45）。すなわち、論理空間上では既存のフリー・メモリ（プール0）の後に連続して該増設メモリ分のフリー・メモリ（プール0）を置くようにアドレス変換テーブルの内容を変更する（図3）。最後に、増減設中指示フラグを元に戻す（S46）。

【0039】以上の処理により、他のファイルを変更することなく物理資源を増設することができる。図5は、一実施例の物理資源の減設方法の説明図である。

【0040】OSファイルやアプリケーション・ファイル1～n、フリー・メモリ（プール0）が論理空間および物理空間上で同図（a）のように割りつけられており、フリー・メモリ（プール0）の一部を減設するものとする。

【0041】減設の場合には、減設対象部分が使用中か否かが問題になる。使用中の場合には、その使用中の部分の内容をフリー・メモリ（プール0）中の未使用領域にコピーしたうえで減設対象部分を取り除く必要がある（同図（b））。

【0042】図6は、一実施例の物理資源減設のフローチャートである。まず、増減設中であることを示すフラグ（通常運転中／増減設中指示フラグ）290を立てる（S61）。そして、OSの管理データを別のメモリ領域に退避する（S62）。

【0043】次に、OSの管理データを変更し、減設対象部分を外すように設定する（S63）。そして、減設対象部分を使用しているものがあるか否かを判断するた

めに、監視タスクを起動する（S64）。監視タスクは、稼働中の他のタスクを妨げることがないように実行レベルの比較的低いタスクとする。監視タスクは、まず、アドレス変換テーブルの物理アドレスをサーチし

（S641）、減設対象部分の物理アドレスがあるか否かを調べる（S642）。ない場合（N）には監視タスクの処理を完了し、元のプログラムに戻る。減設対象部分の物理アドレスがあった場合（Y）には、該物理アドレスの内容をフリー・メモリ（プール0）内の未使用領域にコピーする（S643）。そして、アドレス変換テーブルの該物理アドレスの部分のコピーした領域の物理アドレスに書き換える（S644）。これにより、減設対象部分にあった使用中のプログラムやデータは、フリー・メモリ（プール0）の他の領域にコピーされ、ユーザはこれを意識せずにそのまま使い続けることができる。

【0044】次に、減設対象部分についてハードウェア実装表を変更し（S65）、その後、実際に減設対象部分のハードウェアを減設する（S66）。最後に増減設中指示フラグを元に戻して（S67）、減設処理を完了する。

【0045】以上の処理により、他のファイルを変更することなく物理資源を増設することができる。図7は、一実施例の論理資源の増設方法の説明図である。

【0046】OSファイル等のファイル、および、論理資源（OSが提供するサービスの情報を管理するためのテーブルであるリソース）が論理空間および物理空間上で同図（a）のように割りつけられており、10個分のリソース（リソースn+5～n+14）を増設するものとする。増設前の時点では、1～n+4のリソースが同図（a）のようにアイドルチェーンを形成している。

【0047】本実施例の論理資源の増設方法では、従来のリソース域の後に、予約リソース域として、論理資源増設用の領域を予め確保しておき、その部分で論理資源の増設を行なうことが特徴である。

【0048】図8は、一実施例の物理資源増設のフローチャートである。まず、増設に必要な、例えば、ここでは10個分のリソースに必要なメモリ領域を論理空間内および物理空間内に確保し、アドレス変換テーブル上で、論理空間での位置と物理空間での位置を割りつける（S81）。

【0049】その後、増設したリソース（ここでは10個のリソース）を論理資源としてアイドルチェーンにつなぐ（S82）。つまり、n+4番目のリソースにn+5番目のリソースがつながるように、また、n+5番目以降、n+14までのリソースが順につながるように各リソース内のポインタを設定し、n+14番目のリソースからは先頭に戻るようにポインタを設定することにより、新たなアイドルチェーンを作成する（図7（b））。

【0050】以上の処理により、論理資源の増設処理は完了する。最後に、論理資源の減設方法について説明する。図9は、一実施例の論理資源の減設方法の説明図である。同図(a)は、減設前の論理空間および物理空間、リソースのアイドルチェーンの構成である。

【0051】減設前、リソース域に $n+14$ 個のリソースがあり、そのうちの10個($n+5 \sim n+14$ 番目)のリソースを減設対象とするものとする。図10は、一実施例の論理資源の減設方法のフローチャートである。

【0052】まず、既存のアイドルチェーン(図9(a))から減設対象のリソースをはずす(S101)。すなわち、例えば、 $n+4$ 番目のリソースから $n+5$ 番目のリソースにつながっていたポインタをはずし、 $n+4$ 番目のリソースと先頭とがつながるようにポインタを設定しなおす。次に、アイドルチェーンからははずした $n+5 \sim n+14$ 番目のリソース分のメモリ領域を予約リソース域として、アドレス変換テーブル上で割りつける(S102)。

【0053】以上の処理により、他のファイルに影響を与えることなく論理資源の減設が完了する。

【0054】

【発明の効果】本発明によれば、OSが管理する物理資源および論理資源の増減設を、OSやアプリケーション等のファイルの変更や、OSの初期設定(立ち上げ)をせずに済むようになり、物理資源および論理資源の増減設処理の処理効率が向上することが可能になる。

【図面の簡単な説明】

【図1】本発明の機能ブロック図である。

【図2】一実施例のシステム構成図である。

【図3】一実施例の物理資源の増設方法の説明図である。

【図4】一実施例の物理資源増設方法のフローチャートである。

【図5】一実施例の物理資源の減設方法の説明図である。

【図6】一実施例の物理資源減設方法のフローチャート

である。

【図7】一実施例の論理資源の増設方法の説明図である。

【図8】一実施例の論理資源増設方法のフローチャートである。

【図9】一実施例の論理資源の減設方法の説明図である。

【図10】一実施例の論理資源減設方法のフローチャートである。

【図11】従来の物理資源の増設方法の説明図である。

【図12】従来の物理資源の減設方法の説明図である。

【図13】従来の物理資源増減設方法のフローチャートである。

【図14】従来の論理資源の増設方法の説明図(その1)である。

【図15】従来の論理資源の増設方法の説明図(その2)である。

【図16】従来の論理資源の減設方法の説明図(その1)である。

【図17】従来の論理資源の減設方法の説明図(その2)である。

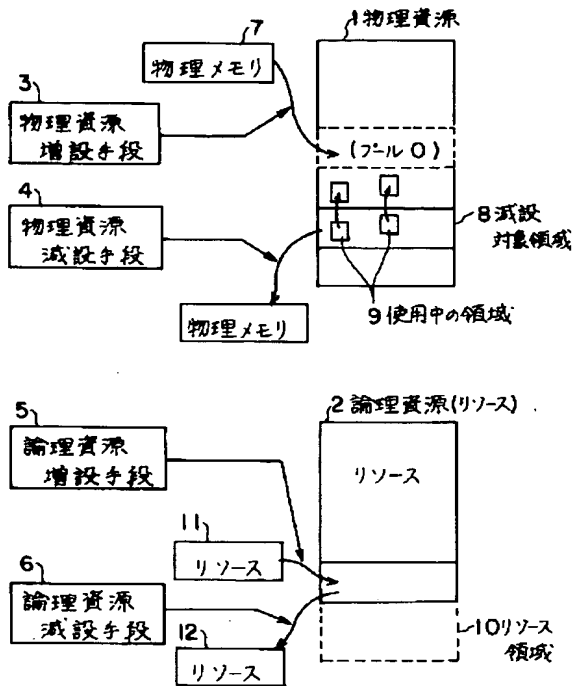
【図18】従来の論理資源増減設方法のフローチャートである。

【符号の説明】

1	物理資源
2	論理資源
3	物理資源増設手段
4	物理資源減設手段
5	論理資源増設手段
6	論理資源減設手段
7	物理メモリ
8	減設対象領域
9	使用中の領域
10	リソース領域
11	リソース
12	リソース

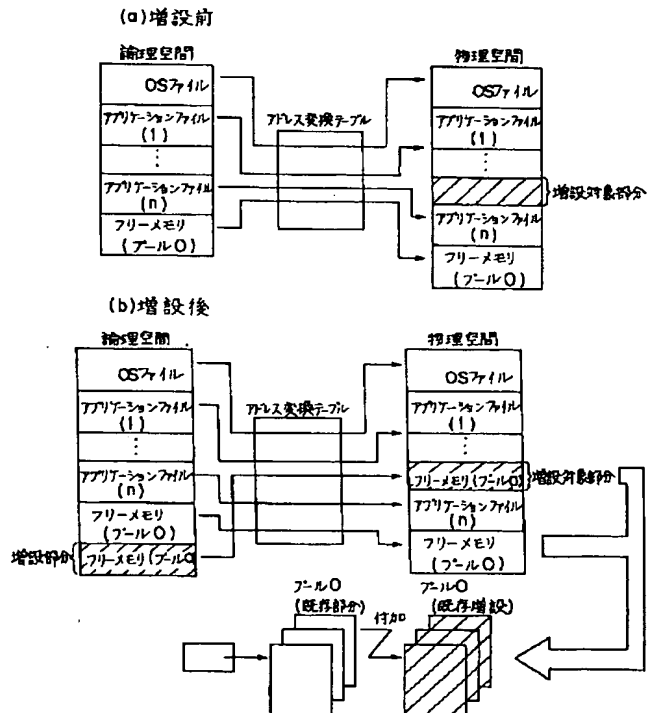
【図 1】

本発明の機能ブロック図



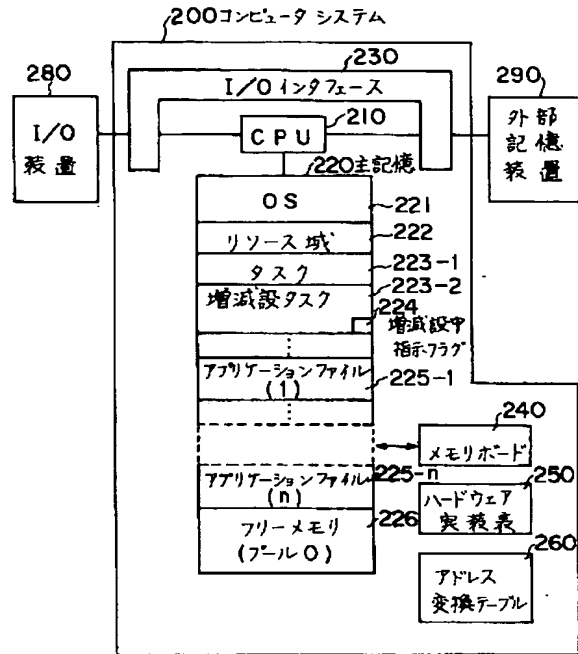
【図 3】

一実施例の物理資源の増設方法の説明図



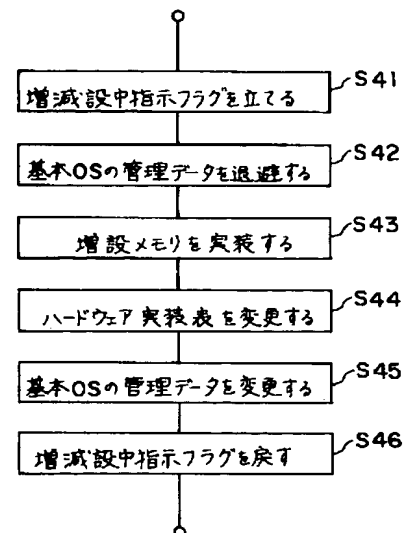
【図 2】

一実施例のシステム構成図



【図 4】

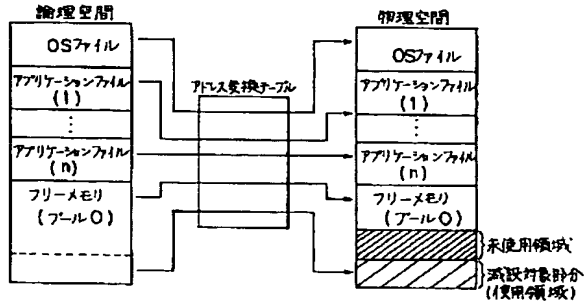
一実施例の物理資源増設方法のフローチャート



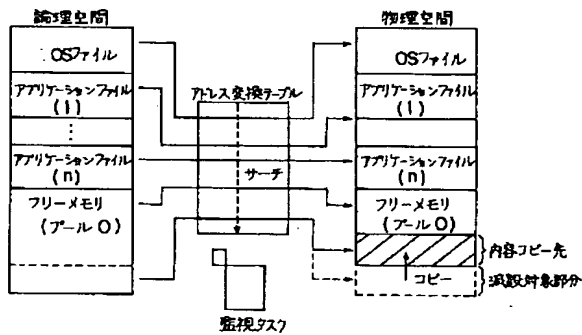
【図 5】

一実施例の物理資源の減設方法説明図

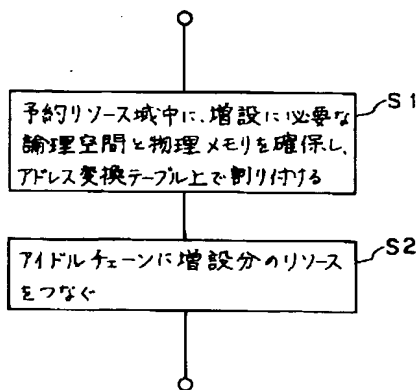
(a) 減設前



(b) 減設後

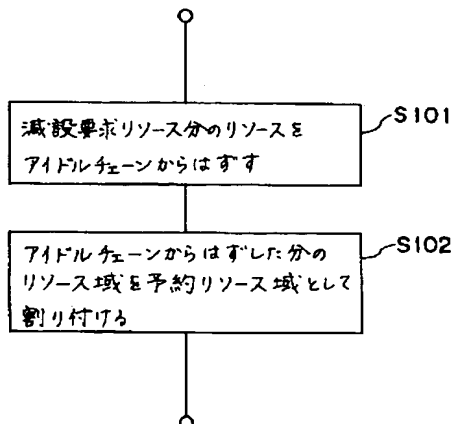


【図 8】

一実施例の論理資源の増設方法の
フローチャート

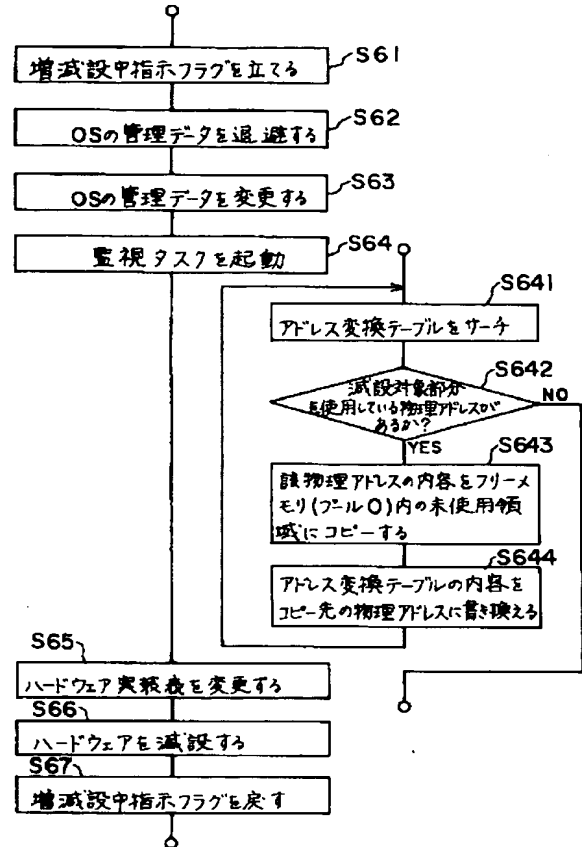
【図 10】

一実施例の論理資源減設のフローチャート

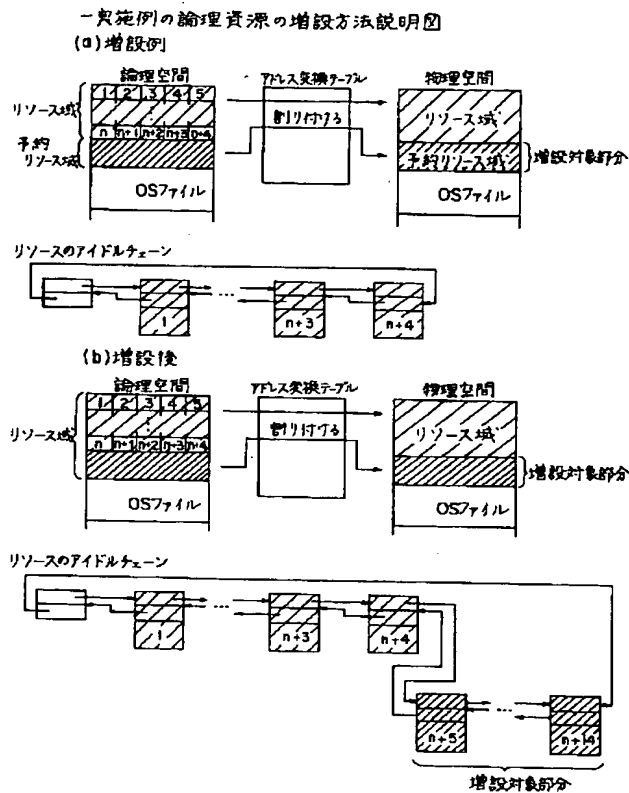


【図 6】

一実施例の物理資源減設方法のフローチャート

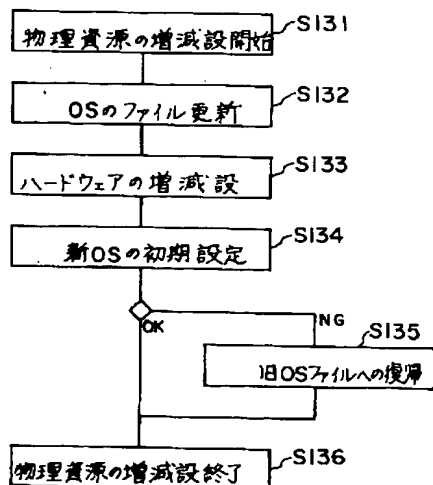


【図 7】

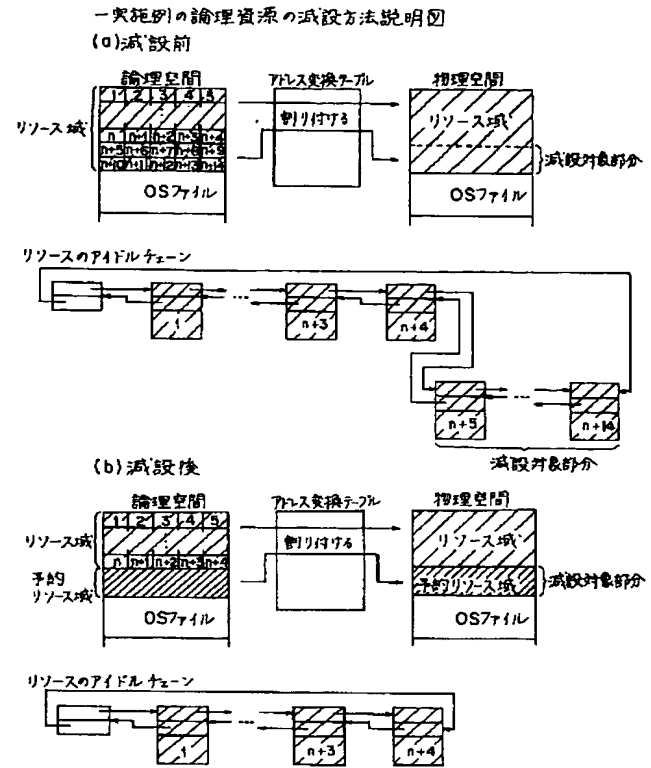


【図 13】

従来の物理資源増減設方法のフローチャート

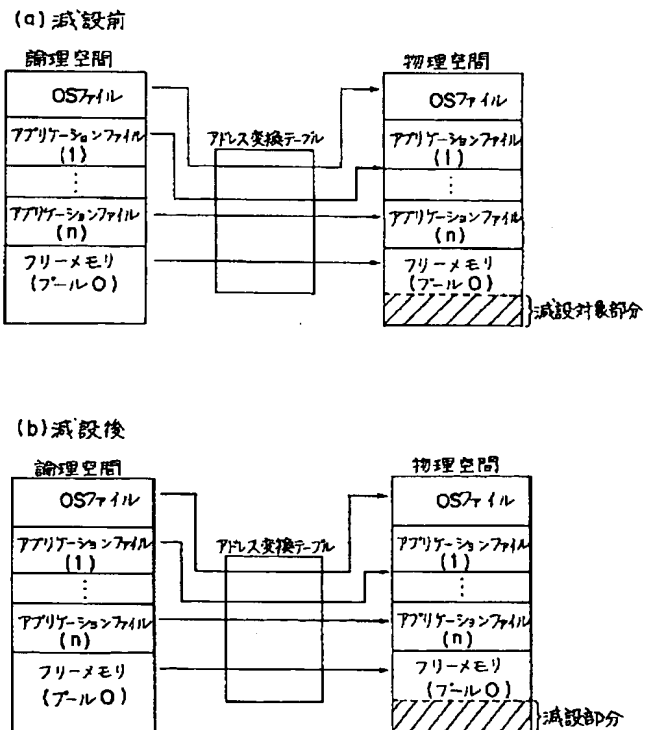


【図 9】



【図 12】

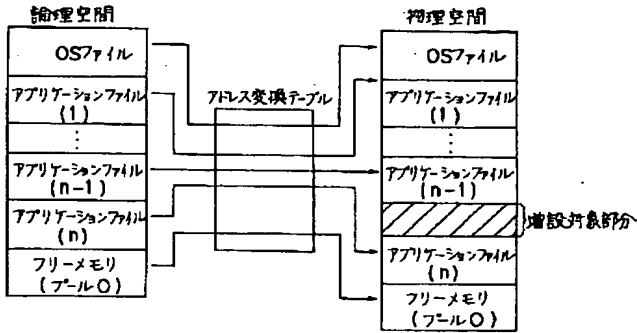
従来の物理資源の減設方法の説明図



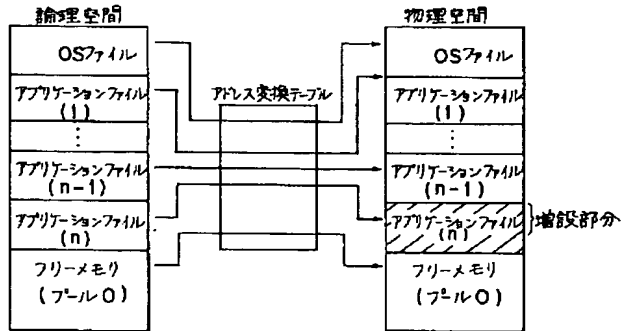
【図 11】

従来の物理資源の増設方法の説明図

(a)増設前



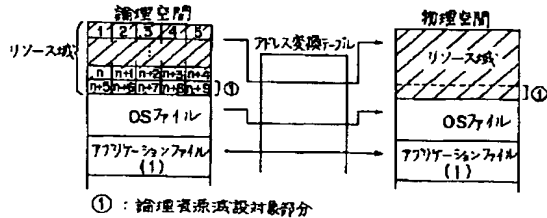
(b)増設後



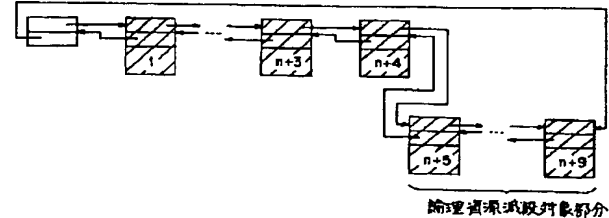
【図 16】

従来の論理資源減設方法の説明図 (その1)

(a)減設前



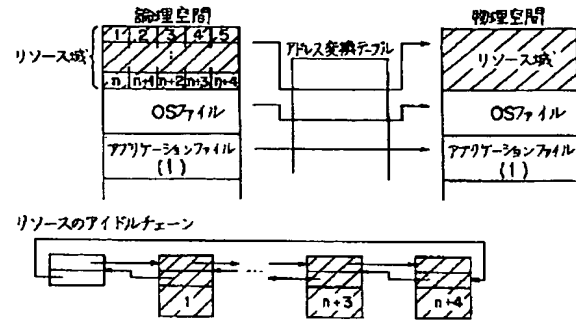
リソースのアイドルチェーン



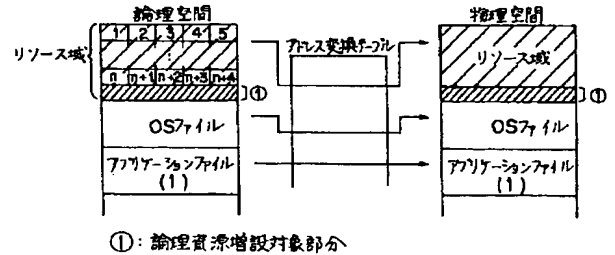
【図 14】

従来の論理資源増設方法の説明図 (その1)

(a)増設前

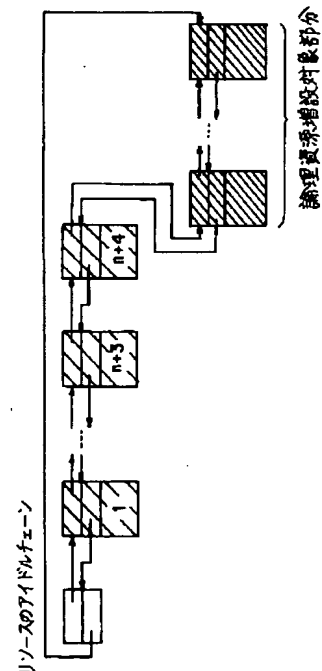


(b)増設後



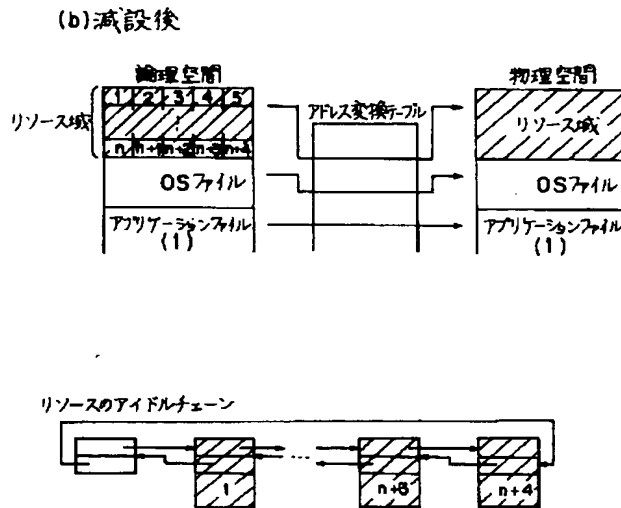
【図 15】

従来の論理資源増設方法の説明図 (その2)



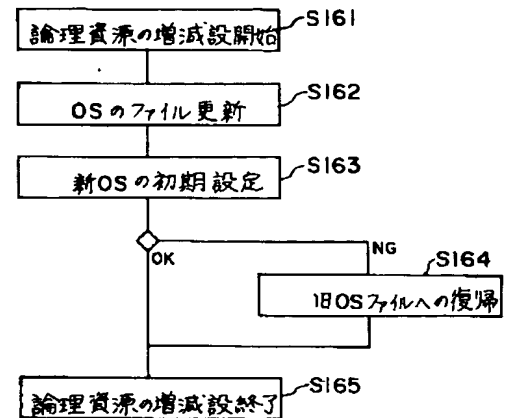
【図 17】

従来の論理資源減設方法の説明図(その2)



【図 18】

従来の論理資源増減設方法のフローチャート



フロントページの続き

(72)発明者 星合 隆成
 東京都千代田区内幸町1丁目1番6号 日
 本電信電話株式会社内